APPLICATION SECURITY TESTING

MYTH, MISTAKES AND NEWS

OCTOBER 25, 2019

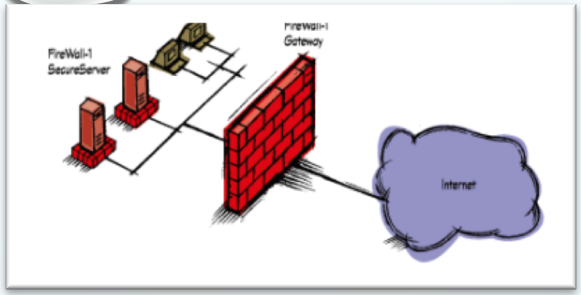
TRIANGLE INFOSECON

JAMIE YU

HTTPS://WWW.LINKEDIN.COM/IN/JAMIEY

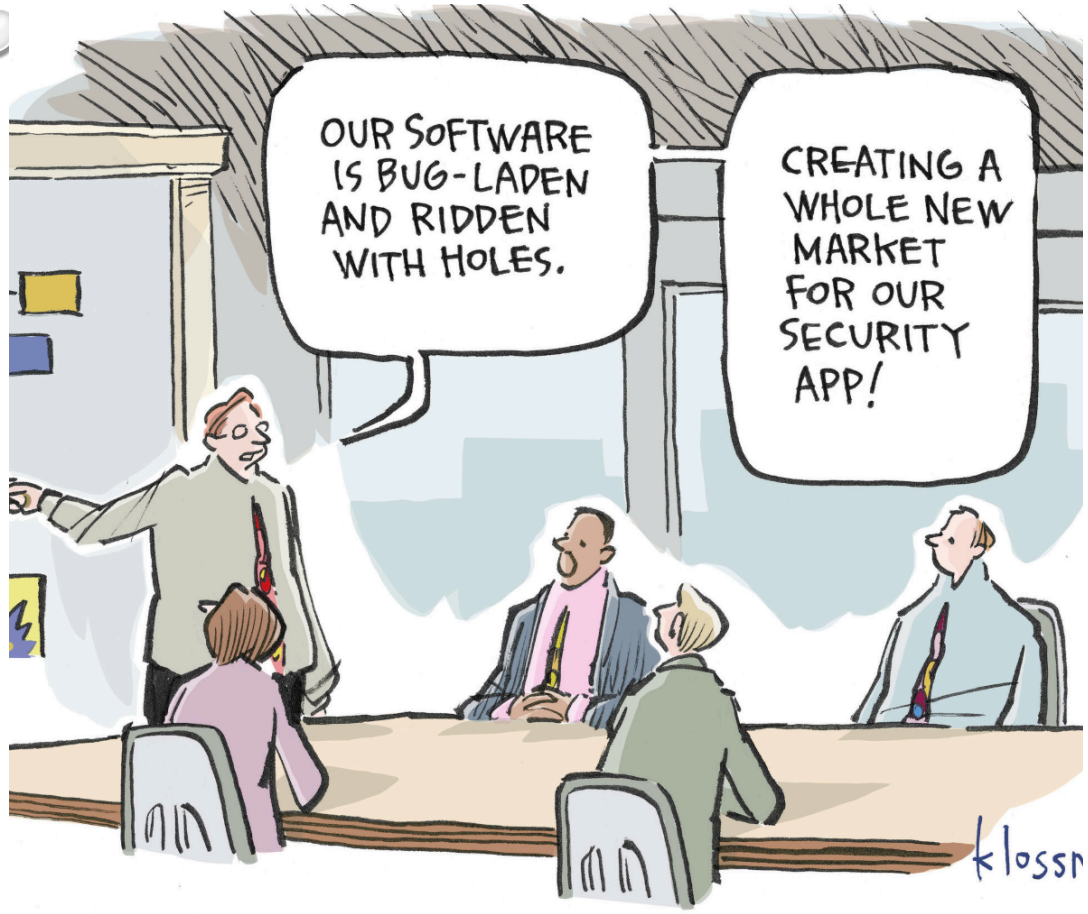
AGENDA

- INTRODUCTION
- SCA
- IAST
- RASP
- CWE
- SECURE WEB APPLICATION
- SECURE DEVOPS



INTRODUCTION

cisco


OUR SOFTWARE IS BUG-LADEN AND RIDDEN WITH HOLES.
CREATING A WHOLE NEW MARKET FOR OUR SECURITY APP!
klossn

APPLICATION SECURITY TOOLS

- SAST
- DAST
- FUZZING

- PEN TEST / BUG BOUNTY
- WAF

- SCA
- IAST

- RASP

THIS DATA BREACH LEAKED SOCIAL SECURITY NUMBERS OF 143 MILLION AMERICANS.

OPEN SOURCE CODE COMPONENTS, FRAMEWORK, PLUG-INS, LIBRARIES

SOFTWARE COMPOSITION ANALYSIS (SCA)

In-depth visibility into the third-party components in application code including open source code

Vulnerability tracking to reduce risk: CVEs, versions, license details, vulnerabilities

Free from false positives

Integration in DevSecOps process

Supports most-popular languages including Jave, C#, .NET, JavaScript, and Obj-C, etc.

SAST

- White Box Security Testing
- Requires Source Code
- Finds Vulnerabilities Earlier
- Less Expensive To Fix
- Supports All Kinds Of Software

DAST

- Black Box Security Testing
- Requires A Running Application
- Finds Vulnerabilities Later
- More Expensive To Fix
- Can Discover Run-time And Environment-related Issues

Interactive Application Security Testing (IAST)

> By 2019, enterprise IAST adoption will have exceeded 30 percent.
>
> —— Gartner magic quadrant for application security testing, 2017

IAST is designed to address the shortcomings of SAST and DAST by combining elements of both approaches. IAST places an agent within the application and performs all its analysis in the app in real-time and anywhere in the development process IDE, continuous integrated environment, QA or even in production.

IAST IS THE MODERN APPROACH TO DAST

- Immediate feedback — zero scan time - IAST is not a scanner!
- Extensive code coverage
- Provide accurate results for fast triage (reduce false positives)
- Identify vulnerable lines of code
- Integrate into CI/CD and DEVOPS workflows
- Allow for earlier, less costly fixes (shift left in SDLC)

IAST DETECTION EXAMPLES

- IAST can detect non-reflective attacks because it handles the reflection and communicates with an induced DAST whether the attack was successful or not.
- IAST can detect sensitive data stored in the logs while DAST can't.

IAST can detect anything a DAST tool can.
IAST covers many of DAST's weak spots.

RUNTIME APPLICATION SECURITY PROTECTION (RASP)

Protects application in runtime - continuous security checks

Protects vulnerable code or libraries

Responds to live attacks by terminating attacker's session

Integrates with SIEM - send alerts in real-time

Might lead to false sense of security and inadequate secure coding practice by developers

IAST AND RASP - PROTECT THE STACK

- IAST detects vulnerabilities in both custom code and libraries during normal use

Vulnerability Confirmed

- RASP prevents vulnerabilities from being exploited in both custom code and libraries

Exploit Prevented

COMMON WEAKNESS ENUMERATION (CWE)

- From Mitre https://cwe.Mitre.Org/
- Not same as Common Vulnerabilities and Exposures (CVE)
- Is a formal list or dictionary of common software weaknesses that can occur in software's architecture, design, code or implementation that can lead to exploitable security vulnerabilities.
- Example of software weaknesses: buffer overflows, format strings, structure and validity problems, common special element manipulations, channel and path error, handler error, user interface errors, pathname traversal and equivalence errors, authentication errors, resource management errors, insufficient verification of data, code evaluation and injection, and randomness and predictability.

This CVE (vulnerability) in iTunes: https://nvd.nist.gov/vuln/detail/CVE-2017-7160
…is a manifestation of this CWE (weakness): http://cwe.mitre.org/data/definitions/119.html

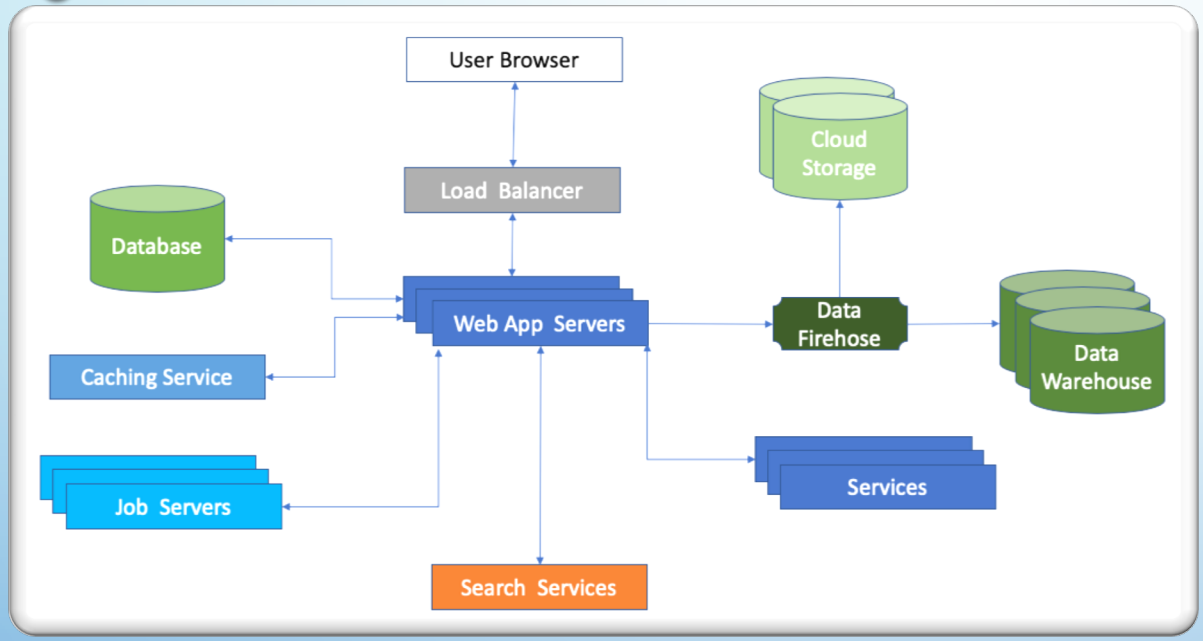
2019 CWE TOP 25 MOST DANGEROUS SOFTWARE ERRORS

Rank	ID	Name	Score
[1]	CWE-119	Improper Restriction of Operations within the Bounds of a Memory Buffer	75.56
[2]	CWE-79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')	45.69
[3]	CWE-20	Improper Input Validation	43.61
[4]	CWE-200	Information Exposure	32.12
[5]	CWE-125	Out-of-bounds Read	26.53
[6]	CWE-89	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')	24.54
[7]	CWE-416	Use After Free	17.94
[8]	CWE-190	Integer Overflow or Wraparound	17.35
[9]	CWE-352	Cross-Site Request Forgery (CSRF)	15.54
[10]	CWE-22	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')	14.10

MITRE has released the 2019 Common Weakness Enumeration (CWE) top 25 most dangerous software errors list. The top 25 is a compilation of the most frequent and critical errors that can lead to serious vulnerabilities in software.

Https://cwe.Mitre.Org/top25/archive/2019/2019_cwe_top25.Html

SECURE WEB APPLICATION



- Authentication
- Authorization
- Session Management
- Data Protection
- Input And Output Handling
- Error Handling And Logging
- Configuration And Operations

EXAMPLE: API AUTHENTICATION

- Weak:
  - API Keys in URL
  - API keys encoded: HTTP Basic Authentication
- Good:
  - API Keys Hashed: HMAC-SHA256
  - API Keys in Header: Oauth2 with MAC
  - TLS with Client Certificate, mTLS

```
GET / HTTP/1.1
Host: localhost:8080
Authorization: Basic aWQ6cGFzcw==
User-Agent: curl/7.43.0
Accept: */*
```

```
$ echo "aWQ6cGFzcw==" | base64 -D
id:pass
```

EXAMPLE: BULK FILE UPLOAD TO AWS S3

Vendor provides a cloud app.
Vendor provides a unique S3 bucket.
Files are encrypted with GPG before upload.

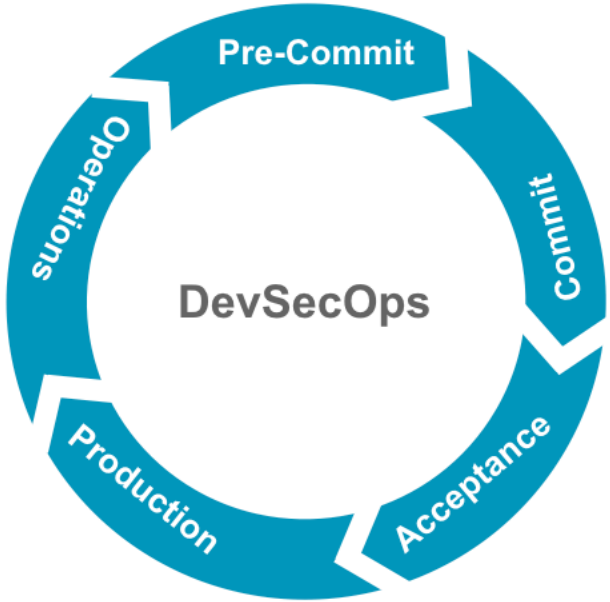
```
curl -T $input_file http://$bucket.s3.amazonaws.com/$path \
    -H "Date: $date" \
    -H "Authorization: AWS $key_id:$sig" \
    -H "Content-Type: $content_type" \
    -H "Content-MD5
```

EXAMPLE: PROTECT THE ENCRYPTION KEY

Files are encrypted with unique key per file per version (AES256).
Keys are stored as clear text in MongoDB.
MongoDB is on the same host as the application.

Pre-Commit:
- Threat Modeling
- Security and Privacy Stories
- IDE Security Plugins
- Pre-Commit Security Hooks
- Secure Coding Standards
- Manual and Peer Reviews

Operations:
- Fault Injection
- Cyber Simulations
- Penetration Testing
- Threat Intelligence
- Continuous Scanning
- Blameless Postmortems
- Continuous Monitoring
- Cloud Monitoring
- Cloud Compliance



Commit (Continuous Integration):
- Static Code Analysis
- Security Unit Tests
- Infrastructure as Code Analysis
- Dependency Management
- Container Security
- Container Hardening

Production (Continuous Deployment):
- Security Smoke Tests
- Configuration Safety Checks
- Secrets Management
- Cloud Secrets Management
- Cloud Security Testing
- Server Hardening
- Host Intrusion Detection System (HIDS)

Acceptance (Continuous Delivery):
- Infrastructure as Code
- Immutable Infrastructure
- Security Scanning
- Cloud Configuration Management
- Security Acceptance Testing
- Infrastructure Compliance Checks

SUMMARY

- INTRO
- SCA
- IAST
- RASP
- CWE
- SECURE WEB APPLICATION
- SECURE DEVOPS

THANK YOU!

- QUESTIONS?